In the Claims:

1. A multiple error correcting code (ECC) mechanism for use with transactions in a computer system, comprising:

an ECC encoder, the ECC encoder applying a first ECC code of a first portion of the transaction and a second ECC code to a second portion of the transaction; and

a decoder that decodes the first and the second ECC codes.

2. The mechanism of claim 1, wherein the first ECC code is a single error correction-double error detection (SEC-DED) code.

3. The mechanism of claim 1, wherein the first portion of the transaction is a header packet and wherein the second portion of the transaction comprises one or more data packets.

4. The mechanism of claim 3, wherein the second ECC code is contained in a last data packet of the one or more data packets.

5. The mechanism of claim 3, wherein the second ECC code comprises a plurality of parity bits and one or more remaining ECC bits, wherein each data packet of the one or more data packets includes one parity bit of the plurality of parity bits, wherein a last data packet of the one or more data packets includes the one or more remaining data bits, and wherein data may be processed without waiting for all of the one or packets to be received if no parity bit errors exist.

6. The mechanism of claim 1, wherein the first portion of the transaction is a first section of a packet and the second portion of the transaction is a second portion of the packet.

7. The mechanism of claim 1, wherein the second ECC code is a SEC-DED code.

8. The mechanism of claim 1, wherein the second ECC code detects a wire failure.

9. The mechanism of claim 1, wherein the transaction further comprises a third portion, and the mechanism further comprises a third ECC code.

10. A method for protecting a transaction in a computer system by using a multiple error correcting code scheme, comprising:

applying a first ECC code to a first portion of the transaction;

applying a second ECC code to a second portion of the transaction;

transmitting the transaction; and

decoding the first and the second ECC codes.

11. The method of claim 10, wherein the first ECC code is a single error correction-double error detection (SEC-DED) code.

12. The method of claim 10, wherein the first portion of the transaction is a header packet and wherein the second portion of the transaction comprises one or more data packets.

13. The method of claim 10, wherein the second portion of the transaction comprises one or more data packets.

14. The method of claim 13, wherein the second ECC code is contained in a last data packet of the one or more data packets.

15. The method of claim 13, wherein the second ECC code comprises a plurality of parity bits and one or more remaining ECC bits, wherein each data packet of the one or more data packets includes one parity bit of the plurality of parity bits, and wherein a last data packet of the one or more data packets includes the one or more remaining data bits.

16. The method of claim 10, wherein the second ECC code is one of a SEC-DED code and a code that detects a wire failure.

17. The method of claim 10, wherein the transaction further comprises a third portion, and the mechanism further comprises a third ECC code.

18. The method of claim 10, further comprising:

defining all transactions and an amount of information to be protected;

defining interconnect widths and packet sizes; and

determining a desired ECC capability.

19. The method of claim 18, further comprising:

listing all frequently-used, multiple packet transactions;

determining a smallest transaction form the list; and

determining if an alternative ECC may be used with the smallest transaction.

20. The method of claim 19, further comprising:

if the alternative ECC may be used, encoding all listed transactions using the alternative ECC; and

if the alternative ECC may not be used:

removing from the list, all transactions whose size equals that of the smallest transaction, and

repeating the process until the list is empty.